Algoritmos e Scratch

APRESENTAÇÃO

Você já se questionou sobre como os jogos virtuais são desenvolvidos? Dependendo do jogo, é possível notar tamanha complexidade, principalmente em alguns aspectos como sons, personagens, cenários, dentre outros. Atualmente, o mercado de desenvolvimento - não apenas de jogos, como também de outros tipos de aplicações - conta com ferramentas e linguagens mais compreensíveis, possibilitando, assim, que qualquer pessoa que se interesse pelo tema possa desenvolver suas próprias aplicações. O ponto crucial, neste caso, está em aprender sobre a lógica de programação.

Nesta Unidade de Aprendizagem, você irá aprender sobre o Scratch, linguagem que busca na representação gráfica trazer os conceitos da lógica de programação que, antes, eram vistos apenas no formato textual. Você irá conhecer, ainda, a interface do Scratch e verá um exemplo prático, de forma a fixar melhor os conceitos que serão aprendidos.

Bons estudos.

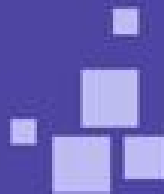
Ao final desta Unidade de Aprendizagem, você deve apresentar os seguintes aprendizados:

- Definir os aspectos básicos do Scratch.
- Descrever a interface do Scratch.
- Usar o Scratch.

INFOGRÁFICO

A linguagem de programação Scratch disponibiliza seus recursos por meio de aspectos visuais, ou seja, ao invés de lidar apenas com linhas de instruções em códigos, o desenvolvedor lida com blocos de instruções já programadas. Para utilizar essa linguagem de maneira proveitosa, é importante conhecer, antes, seu ambiente de programação.

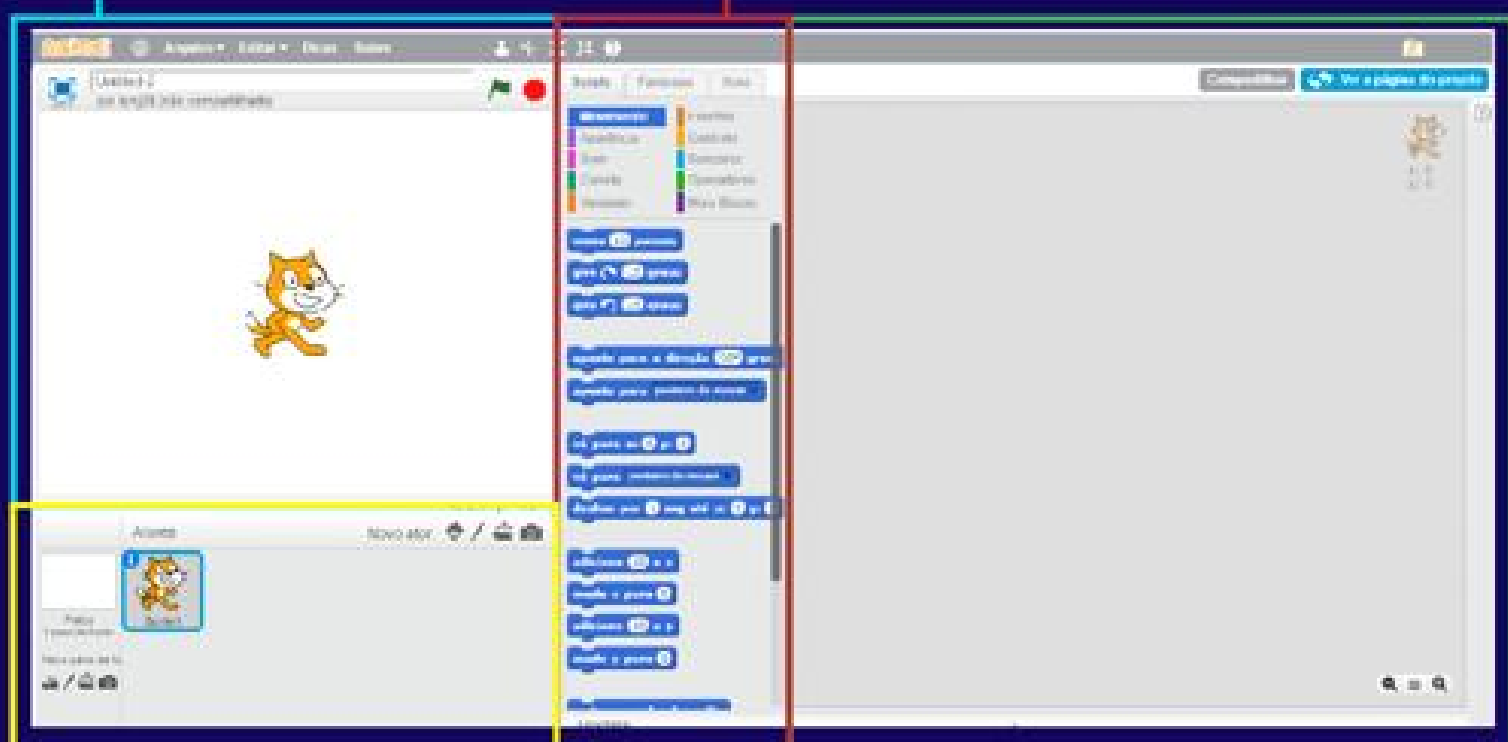
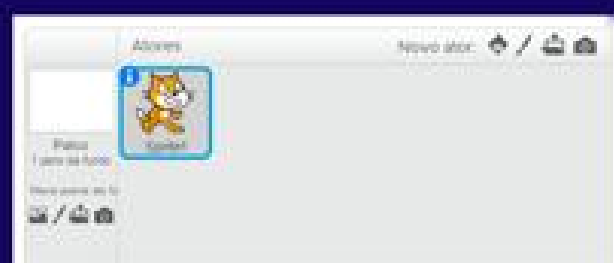
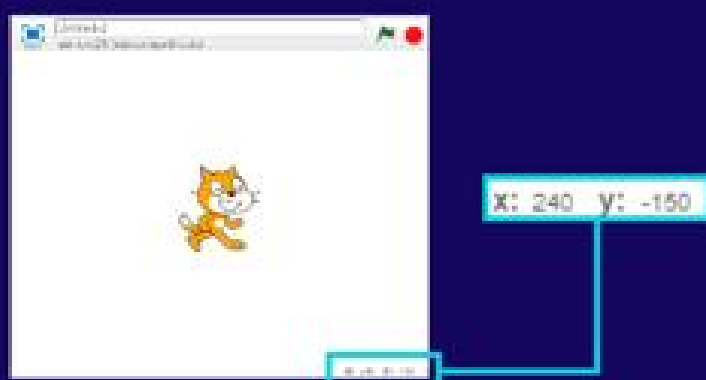
Confira, neste Infográfico, como é composta a interface do Scratch.

INTERFACE DA LINGUAGEM DE PROGRAMAÇÃO SCRATCH

A interface da linguagem de programação Scratch é composta por:

Conheça, a seguir, a descrição de cada uma dessas partes.

STAGE - PALCO

É o local no qual os *sprites* se movem, são desenhados e interagem. O *Stage* tem 480 passos de largura e 360 passos de altura.

O centro tem coordenadas X e Y igual a 0.

SPRIT LIST – LISTA DE *SPRITES*

É o local no qual os nomes e as miniaturas de todos os *sprites* do projeto serão listados.

PALETA DE BLOCOS

No Scratch, os blocos estão divididos em categorias:

- Movimento.
- Aparência.
- Som.
- Caneta.
- Variáveis.
- Eventos.
- Controle.
- Sensores.
- Operadores.
- Mais blocos.

SCRIPTS AREA – ÁREA DE *SCRIPTS*

É o local no qual os blocos de instrução serão inseridos.

CONTEÚDO DO LIVRO

O Instituto de Tecnologia de Massachusetts (MIT) desenvolveu, em 2007, por meio de seu laboratório do Media Lab, a linguagem de programação Scratch. Atualmente, ela está disponível *online* e como aplicação para alguns sistemas operacionais. O Scratch tem como objetivo tornar o desenvolvimento de *software* uma atividade prazerosa, já que a ferramenta traz um ambiente lúdico de desenvolvimento.

No capítulo Algoritmos e Scratch, na obra *Pensamento computacional*, você irá compreender conceitos sobre essa linguagem que busca, de forma lúdica, trazer a compreensão da lógica de programação utilizando blocos de instruções. Além disso, você irá conhecer a interface por meio de exemplos práticos.

PENSAMENTO COMPUTACIONAL

Izabelly Soares de Morais

Algoritmos e Scratch

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Definir os aspectos básicos do Scratch.
- Descrever a interface do Scratch.
- Praticar o Scratch (exemplo prático).

Introdução

Atualmente circulamos por diversos aspectos quando nos submetemos ao processo cognitivo de algo, principalmente se for uma novidade. Dessa forma, a cada dia que passa, os conceitos que antes traziam metodologias tradicionais, hoje, buscam sempre aplicar a dinamicidade no processo de ensino-aprendizagem de um determinado conceito.

Neste capítulo, vamos aprender com a linguagem Scratch, a qual busca na representação gráfica trazer os conceitos da lógica de programação que, antes, eram vistos apenas no formato textual. Com isso, o aprendizado poderá vir a se tornar mais dinâmico e lúdico.

Aspectos básico do Scratch

Todo aplicativo, seja ele qual for, consegue desempenhar suas funcionalidades em razão da execução de um conjunto de instruções, ou seja, a aplicação se comporta conforme foi programada. Até mesmo porque devemos lembrar que o computador não tem discernimento do que seria correto ou não, ou qual deveria ser sua reação após o usuário demonstrar algo ou tentar executar alguma ação.

Para isso, devemos desenvolver maneiras para que essa comunicação entre o usuário e a máquina ocorra, e são as linguagens de programação que permitem que nós, humanos, possamos informar aos recursos computacionais como e quando eles devem desempenhar alguma função.

A linguagem de programação Scratch traz como sintaxe representações gráficas, o objetivo é o de facilitar a compreensão da lógica computacional.

Com isso, jogos, tutoriais e simulações são algumas das ferramentas que podem ser desenvolvidas por meio do uso da linguagem.

Link

Para ter acesso ao Scratch e seu ambiente de desenvolvimento, acesse o link a seguir.

http://scratch.mit.edu.

Conforme o próprio portal da linguagem, o Scratch é um projeto do Lifelong Kindergarten Group do Instituto de Tecnologia de Massachussets (MIT) *Media Lab*. É disponibilizado gratuitamente. A versão atual é a 2. Porém, já está sendo disponibilizada para teste a versão 3 (versão beta). Conforme o MIT, a previsão para lançamento oficial da versão 3 é para dia 2 de janeiro de 2019.

A princípio foi desenvolvido com o intuito de abordar o público de jovens a partir dos 8 anos de idade. Porém, por trazer um ambiente lúdico de programação, é recomendado para todas as idades, tendo em vista, que seus recursos (de multimídia e programação visual) podem ser utilizados para criação de diversos artefatos, desde jogos até animações. Ele está disponível em mais de 40 línguas e em mais de 150 países.

Fique atento

A ferramenta solicitará, caso você ainda não tenha em seu *browser*, a ativação do *plug-in* Adobe Flash Player.

Agora, vamos conhecer um pouco sobre a interface do Scratch.

Conhecendo a interface do Scratch

O próximo passo é nos familiarizarmos com os recursos e a interface do Scratch, até mesmo porque, como já mencionamos, ela traz uma estrutura

um pouco diferente das demais ferramentas utilizadas para desenvolvimento de softwares ou outros artefatos tecnológicos.

Então, após realizar o login e selecionar a opção ***Criar***, iremos nos deparar com a interface de desenvolvimento, de cara, iremos perceber que uma única tela contém diversos painéis e recursos para utilizarmos. A Figura 1 a seguir traz essa divisão.

Figura 1. Interface do ambiente de programação do Scratch.

Podemos observar a primeira interface do Scratch. Quando mencionamos a palavra *primeira*, é no sentido de que não adicionamos nada ainda no nosso projeto. Na barra de menu (Figura 2), temos algumas opções, como a mudança de idioma, que pode ser realizada selecionando o ícone que lembra o globo terrestre. Depois temos as opções relacionadas ao arquivo, que podem ser para salvá-lo, exportá-lo, dentre outras. As próximas opções são as relacionadas à edição. Nela, podemos escolher a configuração visual da interface. O ambiente de programação do Scratch é tão genial que podemos acessar dicas, com tutoriais e documentações, que poderão auxiliar nos nossos projetos. Temos também a opção de saber mais sobre a linguagem.

Figura 2. Barra de menus.

As ferramentas de cursor que ficam ao lado da barra de menus são utilizadas para editar o objeto no palco (Figura 3).

Figura 3. Ferramentas de cursor.

A primeira opção, representada por um ícone que lembra um carimbo, permite a duplicação do ator, a tesoura representa a opção de apagar o ator, as próximas opções são de aumentar e reduzir o ator e, por fim, a opção que traz uma aba de ajuda é referente ao bloco que você posicionar o *mouse*. Essas são as mesmas opções que aparecem ao selecionarmos a opção ***Dica***, na barra de menus (Figura 4).

Figura 4. Ajuda/Dica.

É importante que conheçamos os painéis do ambiente de programação da linguagem e suas funcionalidades. O primeiro painel de que iremos tratar é o ***Palco***, ou, em inglês, ***Stage*** (Figura 5).

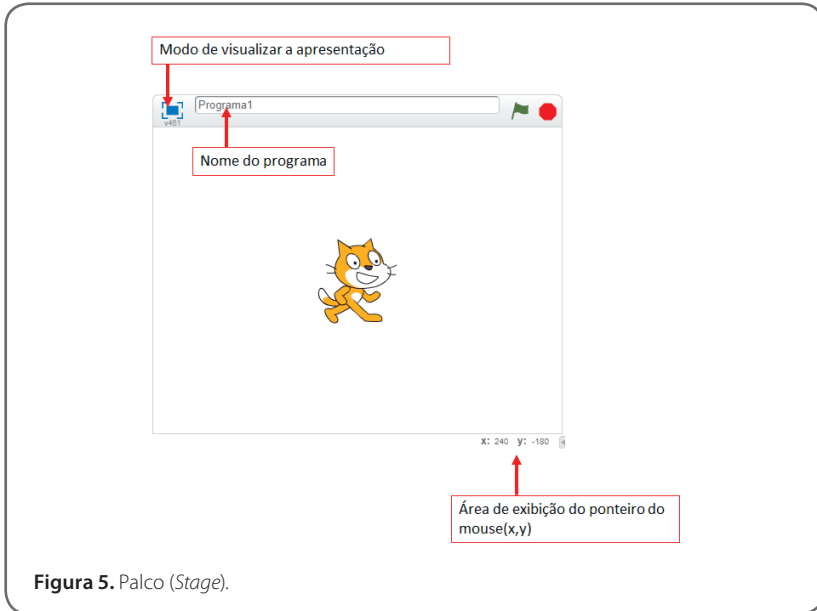
Figura 5. Palco (*Stage*).

Conforme Marji (2014, p. 24), o Palco é o local em que os seus *sprites* (atores) se movem, são desenhados e interagem. O Palco tem 480 passos de largura e 360 passos de altura. O centro do Palco tem coordenada *x* igual a 0 e coordenada *y* igual a 0. As coordenadas (*x*, *y*) de qualquer ponto podem ser encontradas ao mover o cursor do *mouse* até esse ponto e observar os números na *Área de Exibição da Posição* (*x*,*y*) do *mouse*, localizada logo abaixo do Palco, como podemos observar a seguir (Figura 6):

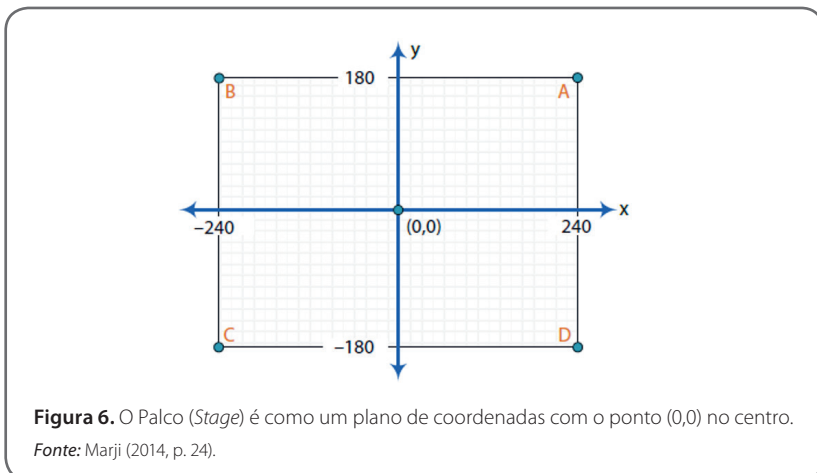
Figura 6. O Palco (*Stage*) é como um plano de coordenadas com o ponto (0,0) no centro.
Fonte: Marji (2014, p. 24).

Para compreender melhor, você pode passar o *mouse* no Palco e observar que os valores de x e y mudam conforme você posiciona o *mouse*. A Figura 7 a seguir nos proporciona uma ideia de como funcionam as coordenadas.

Figura 7. O Palco (*Stage*) — área de exibição.

Para visualizar melhor o posicionamento do *sprite*, ou seja, do ator, que, nos exemplos que estamos mostrando, está sendo representado pelo gato amarelo, você pode selecionar a opção *Modo de apresentação*, que fica no canto superior esquerdo. Dessa forma você pode ampliar a tela (Figura 8).

Figura 8. Ampliar a tela.

Ao ampliar a tela, tudo que estiver no Palco será mostrado em uma outra tela, é como se todas as opções do Palco ficassem ocultas e, para voltar à interface anterior, é só selecionar novamente a opção que a tela voltará ao normal. Ao lado dessa opção, temos um espaço para inserirmos o nome do nosso projeto e, ao lado, a bandeira verde possibilita a inicialização do *sprite*, ou seja, do ator, e com o octógono vermelho, podemos parar o ator. Essa inicialização e finalização pode ser vista como início e término do programa.

Logo abaixo do painel Palco, temos o dos atores, ou seja, é um espaço onde todos os atores do projeto estarão listados (Figura 9).

Figura 9. Lista de atores (*sprites*).

Neste espaço, temos algumas funcionalidades. Até agora, temos um único *sprite* no nosso projeto, já que ainda não estamos planejando executar algo específico. Porém, o interessante é que o nosso ator não precisa necessariamente ser o gato amarelo. No menu superior do painel, ao lado da indicação ***Novo ator***, podemos selecionar outros personagens, como mostra a Figura 10 a seguir.

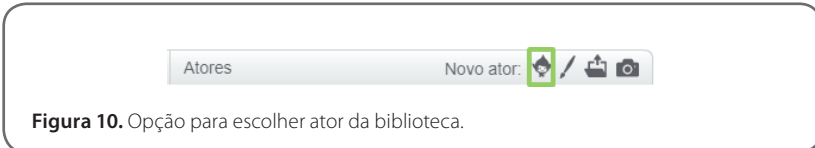



Figura 10. Opção para escolher ator da biblioteca.

Essa possibilidade existe porque o ambiente de programação do Scratch disponibiliza diversos recursos já prontos para que possamos utilizar no nosso projeto. Esses recursos estão armazenados e organizados no que chamamos de bibliotecas. Ao selecionar a opção de escolher ator, nos deparamos com uma gama imensa de recursos (Figura 11).

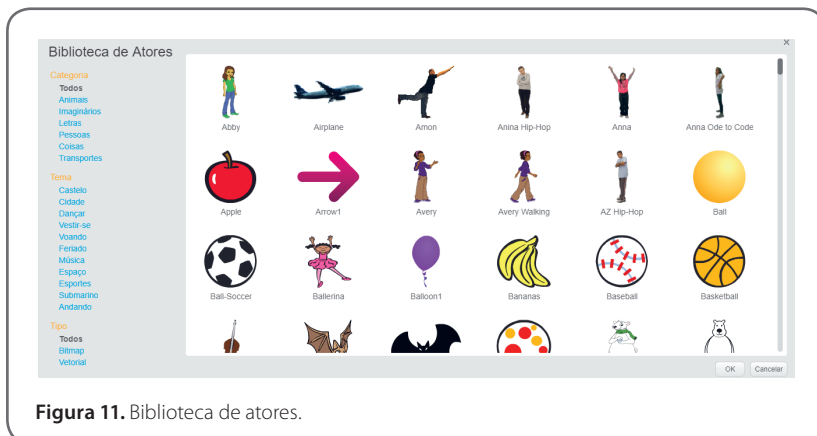



Figura 11. Biblioteca de atores.

No menu ao lado direito, temos: categorias de elementos que podem ser inseridos como animais, imaginários (fantasmas, unicórnios, etc.), letras, pessoas, coisas (diversos objetos aleatórios) e transportes. Abaixo, temos os temas: castelo, cidade, dançar, vestir-se, voando, feriado, música, espaço, esportes, submarino e andando. Posteriormente, temos os tipos: todos, bitmap e vetorial.

Podemos notar que há uma variedade de atores e de contextos diferentes. Lembre-se de que o Scratch 2 possibilita criação de diversos tipos de recursos, como animações, jogos, dentre outros, por isso há toda essa diversidade. Veja a Figura 12.

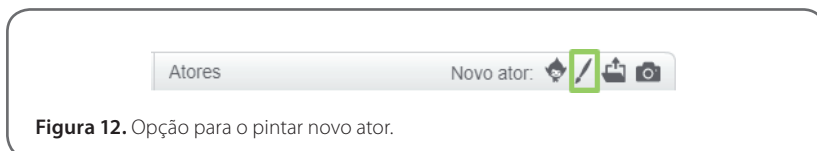



Figura 12. Opção para o pintar novo ator.

A opção de pintar novo ator pode ser chamada também de ***Paint Editor***, tendo em vista que, além do ator, podemos executar outras funcionalidades nesse espaço (Figura 13).

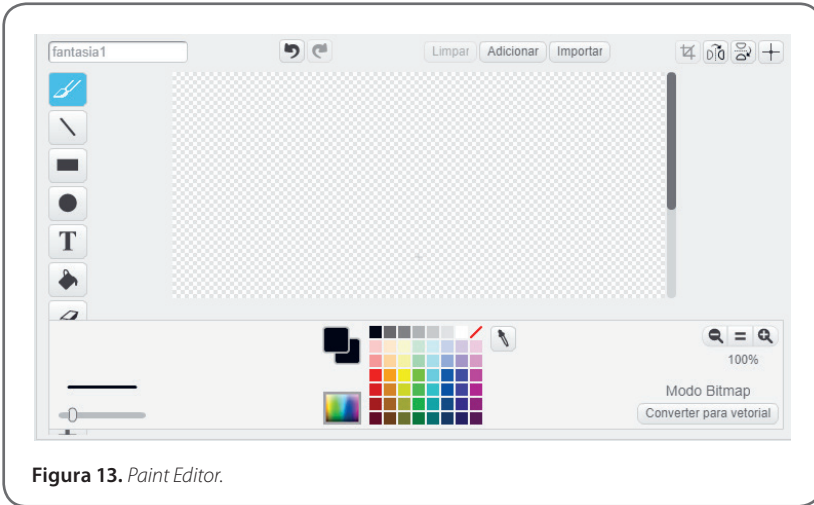



Figura 13. *Paint Editor.*

Uma imagem é composta de pequenos elementos de imagem chamados *pixels*. Esses *pixels* estão dispostos em uma área retangular. As proporções de uma imagem são definidas pela sua largura e altura em *pixels*. No ambiente de programação do Scratch não seria diferente. A seguir, podemos observar em destaque que o ***Paint Editor*** traz dois modos de edição, o bitmap e a opção de converter para vetorial. A tela mostrada na Figura 13 anterior está no modo bitmap. Veja a Figura 14.

Figura 14. Opções do *Paint Editor.*

Observando esse ambiente, podemos rever algumas iconografias a programas que você pode já ter utilizado em seu cotidiano, como o *Paint* e o PowerPoint, que são ferramentas geralmente utilizadas no sistema operacional Windows. Para saber o nome de cada elemento, você pode posicionar o *mouse* em cima e, logo, aparecerá o nome da ferramenta. Veja a Figura 15 a seguir.

Figura 15. Recursos do *Paint Editor — bitmap mode.*

Na Figura 15 anterior, vemos os elementos de edição disponibilizados pelo ambiente de programação e, seguindo a numeração, temos:

1. Pincel — possibilita que o desenvolvedor possa desenhar à mão livre. Podemos, por meio da configuração, alterar as proporções da linha, como largura, cores, dentre outros.
2. Linha — possibilita que seja realizado o desenho de uma linha. Para isso, você deve deslocar e arrastar para uma posição horizontal ou vertical e, por fim, soltar a seleção do *mouse*. Assim como as demais funcionalidades, podemos alterar a largura da linha.
3. Retângulo/quadrado — possibilita que tanto um retângulo quanto um quadrado possam ser criados, tendo em vista que, aplicando os conceitos matemáticos, ambas as figuras geométricas têm quatro lados. Você pode escolher entre as opções de preenchido ou delineado e poderá, também, alterar a espessura de contorno.
4. Elipse — tem as mesmas propriedades do Retângulo, só que, logicamente, para elipse, ou seja, você poderá escolher entre preenchido e delineado e alterar as espessuras do contorno.
5. Caixa de texto — como o nome já diz, é utilizada para inserir um texto. Assim como nos demais recursos, você precisa apenas inserir

no campo de edição. Você poderá alterar, cor, tamanho e fonte deste recurso. O tamanho da fonte você altera apenas aumentando a proporção da caixa de texto.

6. Preencher — você poderá preencher uma região fechada com uma cor sólida ou gradiente. Para isso, poderá escolher um dos estilos de preenchimento (cor sólida, gradiente horizontal, vertical gradiente ou gradiente radial).
7. Borracha — para utilizá-la, você precisa apenas posicioná-la acima do objeto que deseja apagar. Ela funciona como uma borracha que utilizamos no nosso dia a dia, apaga conforme vamos selecionando as partes que gostaríamos que fossem apagadas.
8. Seleção — para utilizá-la selecione uma região retangular e mova-a para um novo local, ou seja, é só você selecionar o que deseja deslocar de posição e arrastar para onde deseja posicionar.
9. Removedor de pano de fundo — você pode selecionar alguma área que deseja remover.
10. Carimbo — para usá-lo você deve selecionar uma região retangular para criar uma cópia dela.

Outras opções existentes ainda nesse campo são alguns botões relacionados a ações gerais, como podemos ver na Figura 16 a seguir.

Figura 16. Botões de controle — *bitmap mode.*

Essas opções estão posicionadas na barra de menu superior ao lado direito. Têm respectivamente as seguintes funcionalidades: cortar alguma área selecionada, inverter o conteúdo da tela (ou apenas a seleção atual) horizontalmente ou verticalmente e escolher um novo centro de rotação. Não podemos deixar de comentar que, como podemos visualizar na Figura 13 anterior, também temos as opções de edição de cores e de panos de fundo dos objetos criados. Agora, vamos conhecer os recursos que teremos disponíveis, caso selecionemos a opção do de modo vetorial (Figura 17).

Figura 17. Opções da conversão para vetorial.

1. Seleção — é uma ferramenta utilizada para mover, redimensionar ou girar objetos.
2. Remodelar — é uma ferramenta utilizada para alterar a forma de um objeto. Para isso, é só selecionar o objeto que você deseja reformular e um conjunto de pontos de controle aparecerão no perímetro do objeto.
3. Lápis — é semelhante à ferramenta pincel do modo *bitmap* que citamos anteriormente. Para usá-la, mantenha o cursor do *mouse* pressionado e mova-o para rabiscar no espaço de edição.
4. Linha — uma linha é definida pelos dois pontos de controle em suas duas extremidades. Para desenhar uma linha, clique no mouse onde deseja que a linha inicie e arraste-o.
5. Retângulo — também tem o funcionamento semelhante ao do modo bitmap que citamos anteriormente.
6. Elipse — também tem o funcionamento semelhante ao do modo bitmap que citamos anteriormente.
7. Caixa de texto — utilizados respectivamente para inserir um texto, preencher algum objeto e duplicar objetos.
8. Preenchimento — utilizados respectivamente para inserir um texto, preencher algum objeto e duplicar objetos.

9. Carimbo — utilizados respectivamente para inserir um texto, preencher algum objeto e duplicar objetos.

Veja a seguir a Figura 18.

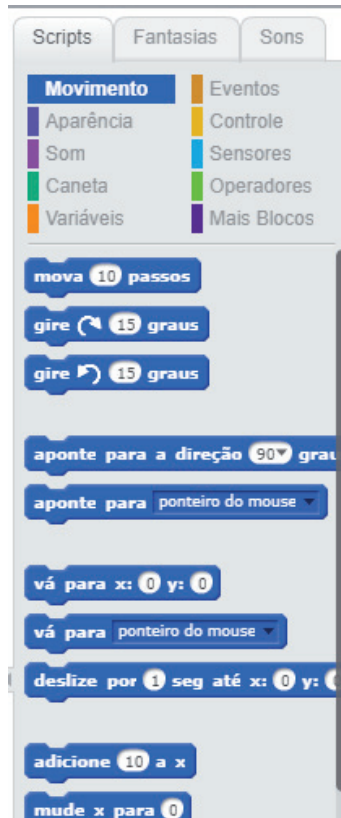



Figura 18. Abas de opções.

Nessas abas vistas na Figura 18 anterior, temos algumas opções, dentre elas os *scripts*, que, se traduzirmos para o português, significa **roteiro**. Aqui, esse roteiro traz na verdade blocos de instruções, ou podemos visualizá-los como roteiros de instruções. Observe na Figura 18 anterior que os roteiros trazem funcionalidades separadas por temas, como: movimento, aparência, som, caneta, variáveis, eventos, controle, sensores, operadores e mais blocos (em que você poderá criar um bloco ou adicionar alguma extensão). Ao lado

dos roteiros (*scripts*), temos as fantasias, que trazem os quadros de animações que cada autor tem. Esse espaço é o mesmo que, quando selecionamos na lista de atores, a opção de pintar novo ator. Temos também a aba de sons, como mostra a Figura 19 a seguir.

Figura 19. Aba para inserção de sons.

Nesta aba, temos a opção de inserir sons e associá-los aos objetos que vamos inserindo no *layout* de nosso projeto.

Agora que já aprendemos um pouco sobre os recursos do ambiente de programação Scratch, vamos colocar em prática alguns ensinamentos já obtidos.

Exemplo prático utilizando o Scratch

Aqui, veremos um exemplo prático de como podemos utilizar a linguagem visual de programação Scratch e, consequentemente, seu ambiente de programação. Lembrando que, para acessá-lo, você deve seguir as instruções ditas lá no início do nosso material. Nosso objetivo, aqui, será simples, vamos fazer com que o nosso ator se desloque da esquerda para direita do palco e, quando ele chegar na borda, ele volte, da direita para esquerda. Os passos iniciais são importantes para que brevemente possamos desenvolver projetos com maior complexidade.

Ao acessarmos o ambiente de programação, nos deparamos com um *sprite* (ator) já criado. Esse ator é representado por um gato amarelo (Figura 20), mas lembre-se de que você pode escolher outro autor.

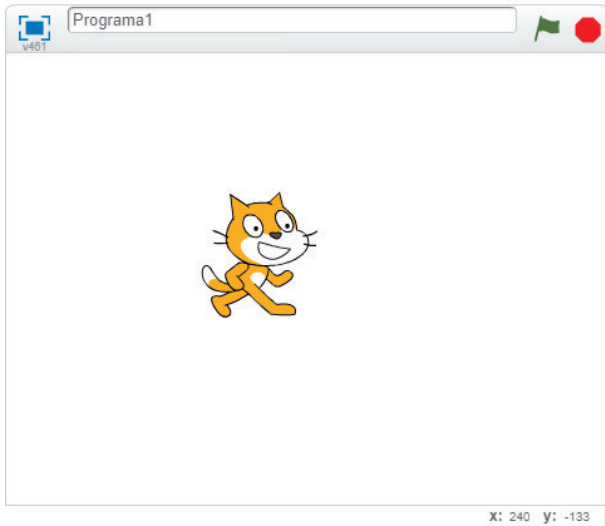



Figura 20. Palco com um ator.

Ao selecionarmos, nos deparamos com as funcionalidades disponíveis para serem utilizadas com esse ator, como mostra a Figura 21 a seguir, na qual notamos que o ator tem duas possíveis “posturas”.

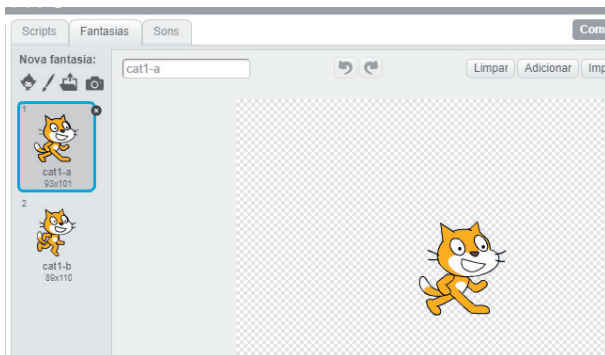



Figura 21. Quadros disponíveis para o ator selecionado.

No nosso exemplo prático, vamos fazer uma alteração inicial, que será no pano de fundo. Para isso, selecione, no painel no qual constam as listas dos atores, a opção de pano de fundo, como mostra a Figura 22 a seguir.

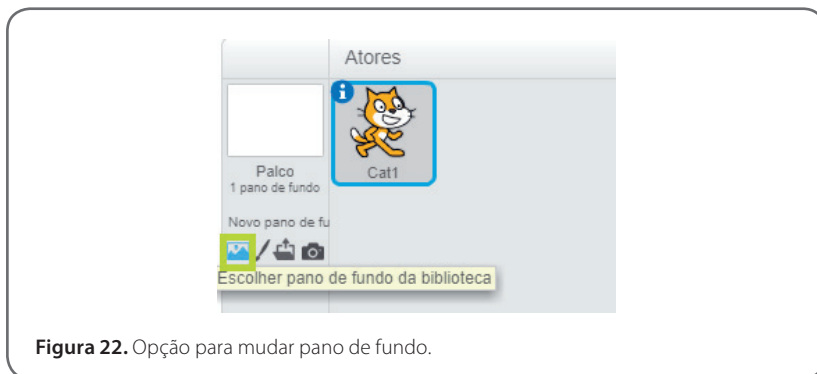



Figura 22. Opção para mudar pano de fundo.

Após esta seleção, podemos notar que diversas opções de panos de fundo são disponibilizadas, lembrando que o mesmo ocorre quando você seleciona opção de escolher ator da biblioteca. Vamos escolher essa opção, porém, ao selecionar o quadro branco, acima da legenda ***Palco***, você se deparará com a seguinte tela (Figura 23):

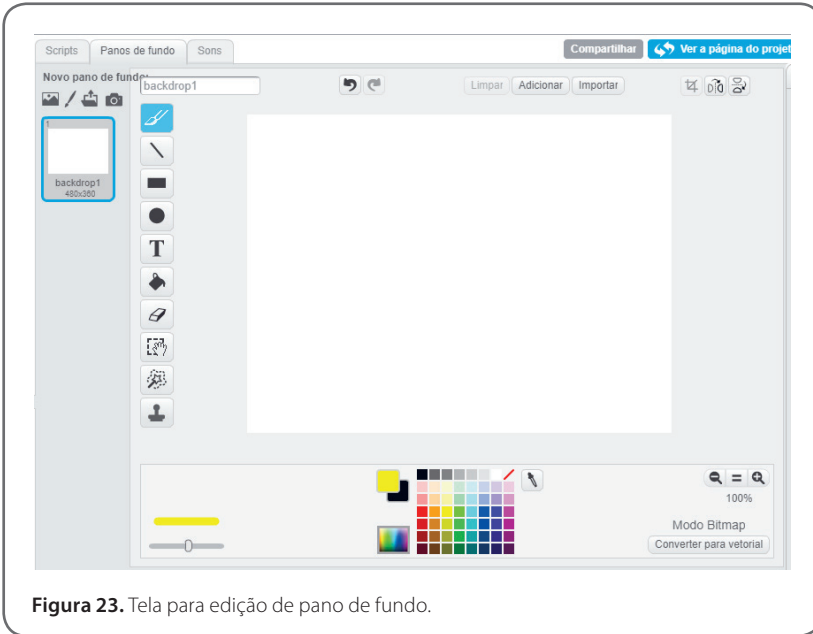



Figura 23. Tela para edição de pano de fundo.

Voltando à nossa opção selecionada, que é a de usar panos de fundo já existentes na biblioteca, iremos nos deparar com muitas opções. Veja na Figura 24 a seguir algumas delas e escolha a que mais lhe agradar. A seleção é feita da mesma forma que podemos escolher o ator: selecione e clique em *ok*.

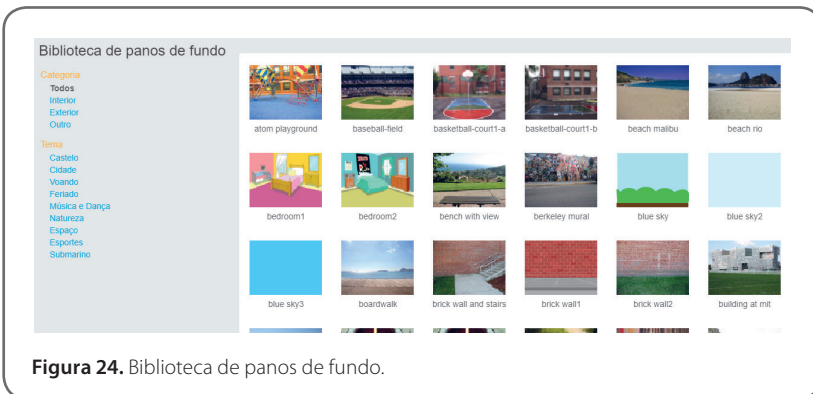



Figura 24. Biblioteca de panos de fundo.

Ao selecionar seu pano de fundo, provavelmente seu ambiente de programação ficará da seguinte forma (Figura 25):

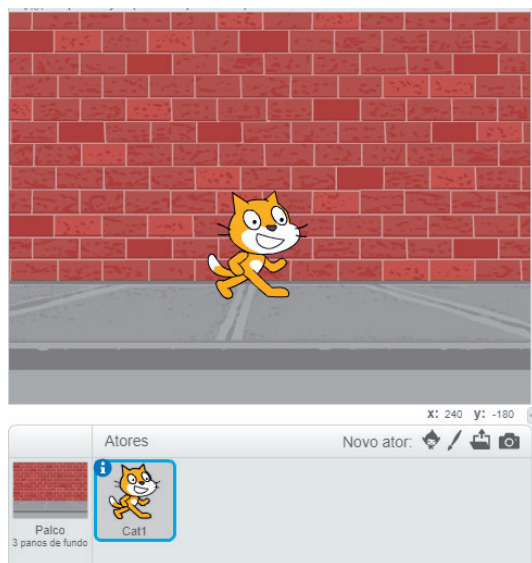



Figura 25. Pano de fundo e ator.

Note que ao modificar o ator de local, ou seja, alterar seu posicionamento no palco, as suas coordenadas também são alteradas. É só você observar os valores de x e y, no canto inferir da área do palco.

Até agora nosso ator, o gato, não fez nenhum movimento, está estático. Agora, vamos dar movimento ao nosso personagem. Aprendemos anteriormente que a linguagem de programação Scratch é lúdica, e é uma linguagem visual, isso quer dizer que, para aplicarmos a programação utilizando essa linguagem e seu ambiente de programação, faremos uso dos roteiros (*scripts*). Já que queremos dar movimento ao personagem, nada mais lógico do que selecionarmos a opção de “movimento” na aba dos *scripts* (Figura 26).

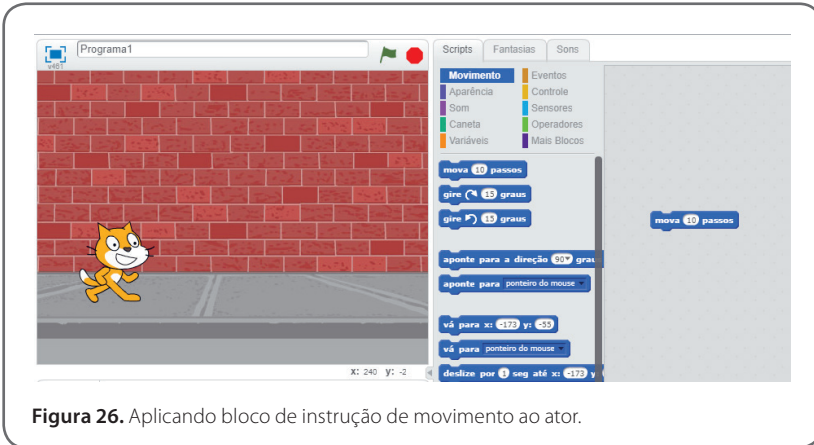



Figura 26. Aplicando bloco de instrução de movimento ao ator.

Posicione seu ator ao lado esquerdo do palco. Posteriormente, na aba de *scripts*, selecione o primeiro bloco de instruções, o que diz ***mova 10 passos***. Note que a ferramenta é extremamente intuitiva, além disso, está disponibilizada totalmente no nosso idioma. Claro que, se você preferir, pode utilizar o idioma que você quiser. Para visualizar o comando em ação, é só você clicar no bloco de instrução. Ao fazer isso, note que a bandeira verde ficará acesa e, quando você para de selecionar, o pentágono vermelho acende. O principal é você perceber que o ato, fica em movimento. No caso, ele dará os 10 passos informados no bloco de instruções para alterar esse valor. Ésó você clicar no campo do número e digitar o novo valor.

Fique atento

O *script* de evento tem como funcionalidade "chamar" a execução de algo na tela. Já o de movimento, como o nome já diz, tem como objetivo movimentar o ator ou algum outro objeto.

Porém, temos outras formas de realizar essa funcionalidade. Na lógica de programação, trabalhamos com algumas estruturas. Essas estruturas facilitam o processo de desenvolvimento de *software*. Elas são estruturas: sequenciais, condicionais e de repetição.

- **Sequenciais**: como o nome já sugere, traz uma sequência. Aqui, essa sequência é aplicada nos blocos de instruções, ou seja, a execução será conforme a ordem que colocarmos. Isso quer dizer que não é adequado que você selecione diversos blocos de forma aleatória, mas sim posicionar apenas o que irá utilizar. Observe também que os blocos de instruções, disponibilizados na aba de *scripts*, têm um "encaixe" que lembram um quebra-cabeça. A seguir, podemos ver alguns blocos de instruções (Figura 27). Para visualizar as outras opções é só acessar a aba de *scripts*.

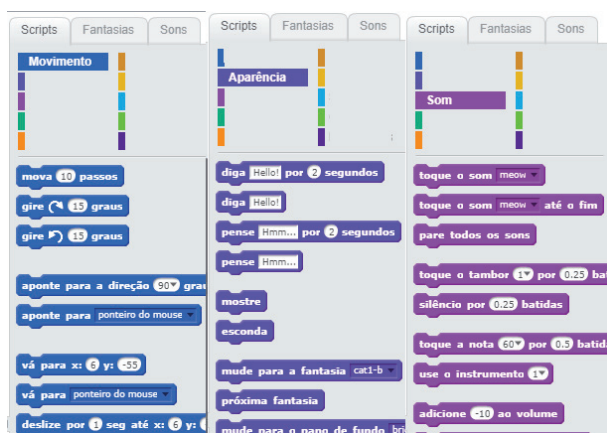



Figura 27. Aba de *scripts*.

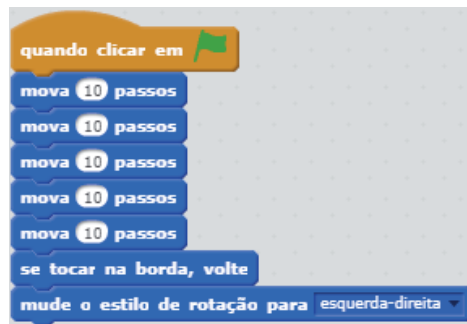
Voltando as nossas estruturas, temos, em seguida, a condicional (na qual alguma ação só será executada após a análise de algumas condições). Aqui na ferramenta é denominada como ***Controle***, inclusive, nesta mesma aba de *scripts*, constam também comandos da estrutura de repetição (são estruturas que possibilitam a repetição de determinadas ações, conforme a quantidade de vezes definida.

Na Figura 26 anterior, podemos observar como nosso *layout* está composto pelo nosso ator (o gato amarelo) e o pano de fundo que inserimos. Agora, vamos começar a inserir os blocos de instruções. Temos várias opções, como mostramos na Figura 27 anterior. O objetivo do nosso exemplo prático é fazer com que nosso personagem se mova de um lado para outro do nosso palco. Veja a Figura 28 a seguir.

Figura 28. Ambiente de programação Scratch.

Para isso, selecionamos na aba *Script* um bloco de comandos ***Eventos***. Quando clicarmos na bandeira verde, as ações serão executadas, no caso, nosso ator irá se mover. Posteriormente, selecionamos um bloco de comandos ***Movimento***, como o ator terá que se mover até a outra borda do palco, podemos fazer de algumas formas, uma delas é repetir o bloco de comandos relacionados ao movimento. Repetimos o bloco de comandos a quantidade de vezes necessárias para que o ator se mova. Os próximos blocos contêm o comando para que, ao tocar na borda, o ator volte, e o bloco seguinte traz a instrução de que, ao tocar na borda, a rotação seja alterada e o ator volte da esquerda para direita. Veja a Figura 29.

quando clicar em
mova 10 passos
mova 10 passos
mova 10 passos
mova 10 passos
mova 10 passos
se tocar na borda, volte
mude o estilo de rotação para esquerda-direita

Figura 29. Blocos de instruções.

Após a inserção dos blocos, é só selecionar a bandeira verde e poderemos visualizar nosso ator se movendo do lado direito para o lado esquerdo, tocando

na borda. Vimos que a ferramenta é riquíssima de detalhes e recursos, dessa forma, não deixe de se aprofundar no que a ferramenta tem a oferecer.

Fique atento

O *Script* de Evento tem como funcionalidade "chamar" a execução de algo na tela.

Referências

MARJI, M. *Aprenda a programar com Scratch*: uma introdução visual à programação com jogos, arte, ciência e matemática. São Paulo: Novatec, 2014.

SCRATCH. Disponível em: <https://scratch.mit.edu/>. Acesso em: 29 nov. 2018.

Leituras recomendadas

BERS, M. U.; RESNICK, M. *The Official ScratchJr Book*: help your kids learn to code. Starch Press, 2015.

JESUS. C.; VASCONCELOS, J. B.; LIMA, R. *Scratch e Kodu*: iniciação à programação no ensino básico. FCA, 2016.

VARELA, H. *Scratch:* um jeito divertido de aprender programação. Casa do Código, 2017.

DICA DO PROFESSOR

O Scratch oferece um conjunto completo de ferramentas multimídia que pode ser usado para criar aplicações variadas. Entre essas ferramentas, está o *Paint Editor*, que oferece diversos recursos, permitindo que o desenvolvedor crie, edite e insira elementos já criados, de forma a adaptá-los da melhor maneira possível ao seu projeto.

Confira, nesta Dica do Professor, alguns recursos do *Paint Editor* e como utilizá-los.

Conteúdo interativo disponível na plataforma de ensino!

EXERCÍCIOS

1) **Ambientes de programação trazem diversos recursos para que o desenvolvedor consiga, com mais facilidade, realizar seus projetos. Sobre o ambiente de programação do Scratch, assinale, a seguir, a alternativa que apresenta as três principais áreas de edição desse ambiente.**

A) Palco, lista de atores e área de *Scripts*.

B) Barra de menus, ferramentas de cursor e mochila.

C) Lista de atores, janela de dicas e paleta de blocos.

D) Abas, palco e mochila.

E) Área de *Scripts*, ferramentas de cursor e abas.

2) **O Scratch é uma linguagem de programação visual cujo objetivo é o de não utilizar a digitação de instruções de comandos. Com base nisso, assinale, a seguir, a alternativa**

que apresenta o conjunto de blocos responsável pela instrução na qual o ator poderá exibir alguma mensagem de texto na tela.

A) Variáveis.

B) Mais blocos.

C) Operadores.

D) Aparência.

E) Controle.

3) **O ambiente da linguagem de programação Scratch possibilita que a criação de novos atores seja realizada de algumas formas. Assinale, a seguir, a alternativa que apresenta os dois tipos de modos disponíveis ao selecionar a opção "pintar novo ator”.**

A) Atores e *Scripts*.

B) Fantasia e atores.

C) Bitmap e vetorial.

D) Vetorial e *Scripts*.

E) Palco e Bitmap.

4) **Entre as janelas presentes no ambiente de programação Scratch, em qual delas é possível verificar a movimentação dos *sprites* (atores)? Assinale, a seguir, a**

alternativa correta.

A) Palco.

B) Lista de atores.

C) *Scripts*.

D) Mochila.

E) Modo vetorial.

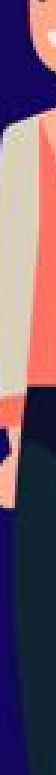
5) **Assinale, a seguir, a alternativa que informa o local no qual é possível visualizar nomes e miniaturas dos projetos criados no ambiente de programação Scratch.**

A) Mochila.

B) Paleta de blocos.

C) Área de *scripts*.

D) Palco.

E) Lista de *sprites*.

NA PRÁTICA

O Scratch oferece um conjunto completo de ferramentas multimídia que pode ser usado para criar aplicações variadas. Ele permite o desenvolvimento de habilidades relacionadas à

resolução de problemas. No entanto, não adianta ter tudo isso e não saber como aproveitar.

Confira, neste Na Prática, algumas outras possibilidades que esse ambiente de programação oferece e um exemplo de como o Scratch pode ser aplicado no dia a dia.

AS POSSIBILIDADES DO AMBIENTE DE PROGRAMAÇÃO DO SCRATCH

Joana ensina informática para crianças na escola municipal de uma certa cidade. Os alunos dela estão na faixa etária entre 10 e 12 anos.

Em uma conferência escolar, Joana conheceu a linguagem de programação Scratch e o seu ambiente de programação. Então, ela teve a ideia de utilizá-la em sala de aula.

Ela explicou para os alunos sobre o ambiente de programação da linguagem e pediu para que eles criassem um personagem e dessem movimento a ele. Joana fez uma única solicitação: que fossem utilizados blocos controle, de forma a evitar que muitas instruções fossem repetidas.

Veja, a seguir, como foi a experiência.

Os alunos começaram a escolher os atores que gostariam de utilizar. Joana utilizou uma borboleta como ator e inseriu um novo pano de fundo para seu projeto, como mostra a imagem a seguir.

Em seguida, Joana indicou quais blocos de instruções os alunos poderiam utilizar para que o ator se movesse entre uma borda e a outra.

Para concluir a atividade, a professora solicitou que os alunos utilizassem os blocos de controle para que não fosse necessário repetir os blocos de instruções de movimento. Após algumas tentativas, a professora resolveu mostrar uma das possibilidades de utilizar o bloco de controle. Ela explicou que, quando a tecla "espaço" for pressionada, a instrução de movimento irá se repetir 10 vezes, ou seja, o ator dará 10 passos, 10 vezes; e, ao tocar na borda, irá voltar, mudando o estilo de rotação. Em todo esse processo, o ator irá tocar o seu som característico, o qual já faz parte dos atores da biblioteca do ambiente de programação da linguagem Scratch. Assim, concluíram a atividade.

```
quando a tecla [espaço] for pressionada
repita (10) vezes
    mova (10) passos
se tocar na borda, volte
mude o estilo de rotação para [esquerda-direita]
toque o som [pop]
```

SAIBA MAIS

Para ampliar o seu conhecimento a respeito desse assunto, veja abaixo as sugestões do professor:

Introdução ao Scratch

Neste vídeo, saiba mais sobre o Scratch, uma linguagem de programação visual, utilizada em diversos contextos e, principalmente, com o intuito de trazer os principais conceitos da lógica de programação.

Conteúdo interativo disponível na plataforma de ensino!

Scratch Fundo Deslizante

A popularidade da ferramenta Scratch é bem mais relevante em países desenvolvidos. Aqui, no Brasil, a ferramenta passa, aos poucos, a fazer parte do processo de ensino e aprendizagem da lógica computacional. Confira, neste vídeo, como mudar o cenário de suas histórias ou de seus jogos nessa linguagem de programação.

Conteúdo interativo disponível na plataforma de ensino!

Tutorial para criar jogos por meio do Scratch

O Scratch oferece a possibilidade de criar diversos tipos de jogos. E essa oportunidade não é apenas para programadores experientes, mas para todos, independentemente da faixa etária. Veja, neste vídeo, os passos primordiais para se criar um jogo.

Conteúdo interativo disponível na plataforma de ensino!